



Variables y memoria

Transcripción

[00:00] Ahora, profundizando más en los conceptos de variables que estamos trabajando y ya refiriéndonos más a los valores en sí con los que estamos trabajando, dónde se guardan estos valores a nivel de memoria y cómo es que Java administra ellos.

[00:18] Para explicar esto un poco mejor vamos a crear nuevamente otra clase y la vamos a llamar EjemploValoresVariables. Le damos finish. Aumentamos el tipo de letra. Nuevamente main, "Ctrl + espacio", seleccionamos el método que queremos, que en este caso es el public static void main string, doble clic y tenemos nuestro método generado.

[00:56] Ahora, a modo de ejercicio digamos que tengo un int número 1 que es igual a 5 y declaro un int número 2 que va a ser igual a 9. Ahora, si yo doy un System.out.println de número 2, "Ctrl + espacio", selecciono la variable que yo quiero que es número 2, obviamente, creo que el resultado ya lo conocemos, va a imprimir un número 9. Perfecto.

[01:37] ¿Qué pasaría si ahora yo digo que número 2 es igual a número 1? Repetimos el experimento, y dado que yo estoy asignando, ahora número 1 vale 5 y yo estoy asignando ese valor a número 2, este número 2 ahora debería ser 5. Vamos a comprobarlo, guardamos, y en efecto ahora hay 5.

[02:12] Ahora, ¿qué pasaría si yo ahora digo que número 1 es igual a 88? ¿Cuánto vale número 2? En algunos lenguajes de programación se basan mucho en lo que son punteros, donde la variable no apunta al valor sino

apunta a la dirección y memoria de esta variable. En el caso de Java no es así. En el caso de Java las variables reciben el valor.

[02:48] Las variables guardan valores y no direcciones ni punteros en la memoria, por lo cual si yo doy un `System.out.print` al número 2, guardamos, ejecutamos, sigue siendo 5. ¿Por qué? Porque número 2 fue asignado con el valor de número 1 que en este momento número 1 vale 5. Pero número 2 no está apuntando a la variable número 1. Número 2 tiene asignado el valor 5.

[03:24] Por lo cual si yo altero después de esto, número 1, una, dos, tres o X veces, número 2 va a seguir teniendo el mismo valor 5. Esto es solamente para que entiendan cómo es que Java administra las variables y la memoria. Java no usa punteros en memorias y en variables sino él trabaja directamente con los valores.

[03:51] Aún estamos en los primeros pasos de Java, entonces poco a poco, conforme vayamos avanzando en los siguientes tópicos, en los siguientes temas, ya van a entender un poco más para qué sirven estos conceptos básicos y van a ver la infinidad de cosas que podemos hacer con Java. Nos vemos en la siguiente clase.